REMARKS

Claims 1-4, 6-10, 17-19, and 21-45 were pending in the Application when last considered by the Examiner. With this Response, Claims 30-32 have been amended, and Claims 29, 36, and 37 have been canceled without prejudice or disclaimer. An unmarked version of all the pending claims is provided above, and a marked version of the claims is provided herewith in an attachment captioned "Version with markings to show changes made."

Claim Rejections - 35 USC § 112

The Examiner rejected Claim 24 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. According to the Examiner, "Claim 24 merely recites 'a communication driver ... comprising a software modem'. It is unclear what elements or functions would encompassed a 'software modem'."

Applicant respectfully disagrees. One of ordinary skill in the art would understand that a "software modem" is a modem which is implemented at least in part in software. Furthermore, details for exemplary embodiments of a software modem are provided throughout the specification and drawing figures of the Application. See, for example, Fig. 2; p.12, lns. 13-15; and p. 13, lns. 29-34. As such, Applicant respectfully requests the Examiner to withdraw the rejection of Claim 24 under 35 U.S.C. § 112, second paragraph.

Claim Rejections - 35 USC § 102

The Examiner rejected Claims 24, 29, 31-33, 35-37, and 43-44 under 35 U.S.C. 102(e) as being anticipated by Suffern et al. (U.S. patent 5,646,983).

Applicant's Claims 29, 36, and 37 have been canceled, thereby rendering moot any rejection of these claims.

Regarding Claim 24, the Examiner states: "Suffern teaches a system for using a host processor to perform modem transmissions functions [modulation and demodulation funtions--see abstract]. Since Suffern's software program provides control and interfaces with the conversion interface adapter 15 [see figure 4], it meets the definition of a communication driver as claimed."

Applicant respectfully disagrees. Suffern et al. does not disclose any sort of "communication driver" as recited in Claim 24. Indeed, the Examiner admits as much in another part of the Office Action, where the Examiner states, "Suffern does not disclose a communication driver." (Office Action, p. 7). As such, Suffern cannot anticipate Applicant's Claim 24.

For at least these reasons, Applicant respectfully requests that the Examiner withdraw the rejection of Claim 24 under 35 U.S.C. 102(e) and to allow this claim. Since Claims 25-28 depend from Claim 24, it is further requested that the Examiner withdraw the rejection of these claims and allow these claims as well.

Applicant's Claims 31 and 32, as amended, now depend from Claim 30 which, as amended, is allowable as discussed below. For at least the reasons discussed below, Applicant respectfully requests the Examiner to withdraw the rejection of Claims 31 and 32 and allow these claims as well.

Claim 33 recites, in part, "determining data received based on a waveform represented by the sampled digital values, and based on a modem protocol, wherein said determining is

performed in a processing unit coupled to the analog to digital converter by a local bus of a computer, the sampled digital values being transferred from the analog to digital converter to the processing unit by the local bus; and providing the data received to an operating system."

Applicant respectfully submits that Suffern et. al does not disclose or teach such limitations. In Suffern et. al, a data demodulation (demod.asm) module 109 functions to demodulate filtered sample values into data. See Col. 7, lns. 22-24. But such data is never provided to any operating system. Instead, the data is provided to a DBOX.ASM module 111 which "provides routines which allow the host computer's screen to provide an oscilliscope-like eye-diagram display useful for monitoring the performance of the system during modem data reception." See Col. 7, lns. 25-29. Indeed, the DBOX ASM module 111 is part of a program that is loaded for execution "by entry of the program's name at the operating system's standard command prompt" (see col. 6, lns. 42-54 and col. 8, lns. 3-5), and as such, is distinct from the operating system. Therefore, Applicant's Claim 33 cannot be anticipated by Suffern et al.

For at least these reasons, Applicant respectfully requests that the Examiner withdraw the rejection of Claim 33 under 35 U.S.C. 102(e). Since Claim 35 depends from Claim 33, it is further requested that the Examiner withdraw the rejection of this claim.

Applicant's Claim 43, in part, recites, "a processor coupled to the local bus and programmed to: determine data received based on a waveform represented by the sampled digital values and based on a modem protocol; and provide the data received to an operating system of the computer." As discussed above with respect to Claim 33, Suffern et. al. does not disclose or teach providing received data to an operating system, and thus, this claim cannot be anticipated by Suffern et al.

For at least these reasons, Applicant respectfully requests that the Examiner withdraw the rejection of Claim 43 under 35 U.S.C. 102(e). Since Claim 44 depends from Claim 43, it is further requested that the Examiner withdraw the rejection of this claim.

Claim Rejections - 35 USC § 103

The Examiner rejected Claims 1-2, 4, 6-9, 17-19, 21-23, 25-28, 30, 33, 35, 38-42, 44, and 45 under 35 U.S.C. 103(a) as being unpatentable over Suffern et al. and further in view of Bailey et al. (U.S. patent 5,644,593).

Applicant respectfully traverses. In rejecting various independent claims (e.g., Claim 1, 33, and 43) the Examiner asserted that "It would have been obvious for one of ordinary skill in the art to uses the driver of Bailey with the device of Suffern because it would have enable the device of Suffern to be used by existing application software written for communicating to a standard serial interface without modification to the application software." Applicant, however, respectfully submits that Suffern et al. and Bailey et al. teach away from their combination.

Bailey et al. describes a problem of the recognized prior art as follows:

> [I] n a typical personal computer, an unbuffered UART receives data one bit at a time until an asynchronously framed byte (8 bits of data, 1 start bit and 1 stop bit) is received. The UART then signals the CPU of the personal computer (via a serial interrupt) to indicate that it has received a byte of data. If the CPU does not service the serial interrupt before the next byte of data is received, the previous byte of data is over written and the UART indicates that an overrun error had occurred. Under ordinary conditions, the data is lost. The only way to avoid losing data is to utilize a higher level protocol or software layer which upon detecting the error can negotiate with the transmitting DTE at the remote end to retransmit the lost data or the block containing the lost data. In spite of these higher level protocols, even a small number of overran errors can significantly degrade the performance the communications link. If the CPU is forced to service a serial interrupt for each byte of data at very high

data rates, the frequency of serial interrupts that will occur can account for a significant amount of the CPU time causing the operating system to grind to a halt or make it so sluggish that it will be impractical.

Bailey et al. Col. 3, lns. 11-32. Thus, Bailey et al. teaches that it is desirable to *reduce* the load on a CPU.

In contrast, Suffern et al. teaches that it is desirable to *increase* the processing load on a CPU, in order to minimize costs. In particular, Suffern et al. describes that the processing of signals for modem functionality can be moved away from a separate modem unit to the microprocessor (CPU) of a host computer, thus reducing the costs for a modem by eliminating the need for separate processors in the modem unit. See Suffern et al., col. 1, lns. 60-67.

Because the stated objectives of Suffern et al. and Bailey et al. are polar opposites, these references cannot be combined.

For at least this reason, Applicant respectfully requests the Examiner to withdraw the rejection of Claims 1-2, 4, 6-9, 17-19, 21-23, 25-28, 30, 33, 35, 38-42, 44, and 45 under 35 U.S.C. 103(a) and to allow claims.

The Examiner rejected Claims 3, 10, and 34 under 35 U.S.C. 103(a) as being unpatentable over the combination Suffern et al. and Bailey et al. and further in view of Gibson et al. (U.S. patent 5,640,594).

As discussed above, Suffern et al. and Bailey et al. teach away from their combination. Furthermore, Claims 3 and 10 depend from Claim, and Claim 34 depends from Claim 33. Claims 1 and 33 should be allowable, as discussed above. For at least these reasons, Applicant respectfully requests the Examiner to withdraw the rejection of Claims 13, 10, and

34 under 35 U.S.C. 103(a) and to allow these claims. Furthermore, Claims 3 and 10 depend from Claim 1, and Claim 34 depends from Claim 33. Claims 1 and 33 should be allowable, as discussed above.

CONCLUSION

For the above reasons, Applicants respectfully request that pending Claims 1-4, 6-10, 17-19, 21-28, 30-35, and 38-45 be allowed and this case passed to issuance. Should the Examiner wish to discuss the Application, it is requested that the Examiner contact the undersigned at (415) 217-6000.

EXPRESS MAIL LABEL NO:

EL 947 823 640 US

Respectfully submitted,

Philip W. Woo

Philip W. Woo
Attorney of Record
Reg. No. 39,880

Version with markings to show changes made

In the Claims

For the convenience of the Examiner, all pending claims, whether amended or not, have been provided. Un-amended claims are shown in italics.

Please amend the claims to conform to the following complete set.

1. *A system comprising:*

a computer having a processing unit, a main memory, and a local bus;

a device coupled to the local bus, wherein the device occupies an I/O slot on the local bus and is accessible at a first set of addresses corresponding to a first communications port, and the device has a register set with an address assignment in the first set of addresses that differs from a standard address assignment of a register set for a UART corresponding to the I/O slot; and

a communications driver executed by the processing unit, the communications driver comprising a UART emulation which in response to an access targeted at a register set of a UART corresponding to the first communication port, converts the access as required for the register set and address assignment of the device.

2. *The system of claim 1, wherein the local bus comprises an ISA bus.*

3. *The system of claim 1, wherein the device coupled to the local bus, further comprises:*

a comparator adapted for receiving a data signal from the local bus;

a pattern generator coupled to the comparator, wherein the pattern generator generates a signal for comparison with the data signal;

a counter operably coupled to the comparator, wherein the counter resets to an initial state following the comparator indicating the data signal is not equal to the pattern signal and advances toward a final state following the comparator indicating the data signal equals the pattern signal; and

a register coupled to the counter and adapted to receive a signal from the local bus, wherein in response to the counter reaching the final state, the register latches from the local bus a value which indicates the base address of the I/O slot occupied by the device.

4. *A method for communicating between a computer and a device having an I/O interface which differs from the I/O interface of a UART, comprising:*

coupling the I/O interface of the device to a local bus in the computer;

allocating in a memory of the computer, storage locations which correspond to registers of a UART;

transmitting a packet formatted for a UART via a communications driver including a UART emulation;

updating a value in the storage locations according to a value in the packet via the UART emulation; and

transmitting information via the local bus between the I/O interface of the device and the allocated storage locations in the memory of the computer.

6. *The method of claim 4, wherein an I/O handler performs the step of said transmitting information by:*

converting a value from the allocated storage to a converted value compatible with the I/O interface of the device; and

writing the converted value to a register in the device via the local bus.

7. *The method of claim 4, wherein said transmitting information further comprises:*

reading values from a register in the device via the local bus; and

updating the storage locations according to the value read.

8. *The method of claim 7, further comprising transmitting from a communications driver to an application information from the storage locations.*

9. *The method of claim 4, further comprising:*

executing on the computer an operating environment which allocates I/O slots on the local bus for UARTs; and

setting a base device address for the device to correspond to one of the I/O slots allocated by the operating environment for the UART.

10. *The method of claim 9, wherein setting the base device address comprises:*

sensing, by the device, of a data signal on the local bus;

comparing the data signal to a signal from a pattern generator in the device;

advancing a state indicator toward a final state in response to the data signal being equal to the signal from the pattern generator;

repeating the steps of sensing, comparing, and advancing until the state indicator reaches the final state; and

setting the base address of the device to a value indicated by a signal on the local bus in response to the state indicator reaching the final state.

17. A host signal processing modem comprising:

a device adapted for connection to a local bus of a host computer, wherein the device occupies an I/O slot on the local bus and is accessible at a first set of addresses, the device having a register set with an address assignment in the first set of addresses that differs from a standard address assignment of a register set for a UART corresponding to the I/O slot; and

a communications driver executable by the host computer, the communication driver comprising a UART emulation, wherein in response to the host computer executing a procedure that targets an access at a register set of a UART, the UART emulation converts the access as required for accessing the register set and address assignment of the device.

18. The modem of claim 17, wherein the procedure that targets an access at the register set of a UART is part of an operating system that the host computer executes.

19. A communication driver executable by a host computer running an operating system that assigns a first port to a UART, the communication driver comprising:

a UART emulation that in response to a procedure requesting access to a register of a UART at a first port, instead accesses storage locations in a memory of the host computer; and

an I/O handler that transfers values between the storage locations and a register set of a non-standard device having an address assignment that differs from

that of a UART, wherein the register set of the non-standard device physically occupies addresses corresponding to the first port.

21. The communication driver of claim 19, further comprising modem software that implements a conversion between data and digital samples representing a signal in accordance with a communication protocol.

22. The communication driver of claim 19 wherein the address of a first storage location corresponds to a line control register, the address of and a second storage location corresponds to a line status register.

23. The host signal processing modem of claim 17 wherein the register set includes a line control register, a line status register and a modem control register.

24. A communication driver executable by a host computer, the communication driver comprising a software modem.

25. The communication driver of claim 24, further comprising software that accesses storage locations in a memory of the host computer in response to a call requesting access to a register of a hardware UART.

26. The communication driver of claim 25, wherein the software modem converts between data and digital samples of waveforms in accordance with a modem protocol.

27. The communication driver of claim 26, further comprising an I/O handler that transfers values between storage locations in the memory of the host computer and a register set of a non-UART chip in a peripheral device of a host computer.

28. The communication driver of claim 27, wherein the peripheral device further comprises an analog-to-digital converter and a digital-to-analog converter.

Please cancel Claim 29 without prejudice or disclaimer.

30. (Amended) [The system of Claim 29 wherein:] <u>A system comprising:</u>

<u>a device comprising an analog to digital converter couplable to a communication medium to receive therefrom an analog communications signal; and</u>

<u>a computer comprising a processing unit coupled to the device, to receive therefrom a plurality of sampled digital values, the processing unit being programmed with a software modem to determine data received, based on a waveform represented by the sample digital values, wherein</u> the processing unit is programmed with an operating system[; and

the operating system supports] <u>for supporting</u> a plurality of applications, at least one of the applications communicating with the software modem in the same manner as with a hardware modem.

31. (Amended) The system of Claim [29] <u>30</u> wherein:

the device generates interrupts; and

the software modem reads a set of sampled digital values from the analog to digital converter in response to an interrupt.

32. (Amended) The system of Claim [29] <u>30</u> wherein:

the device further comprises a digital to analog converter coupled to the communication medium to transmit thereto an analog signal; and

the software modem generates a series of digital values sent to the digital to analog converter for transmission as an analog signal on the communication medium, the analog signal providing a carrier signal and data values formatted according to a standard modem protocol.

33. *A method comprising:*

converting an analog communications signal received from a communication medium into a series of sampled digital values, wherein said act of converting is performed in an analog to digital converter;

determining data received based on a waveform represented by the sampled digital values, and based on a modem protocol, wherein said determining is performed in a processing unit coupled to the analog to digital converter by a local bus of a computer, the sampled digital values being transferred from the analog to digital converter to the processing unit by the local bus; and

providing the data received to an operating system.

34. *The method of Claim 33, wherein:*

the analog to digital converter is a portion of a device that does not comprise a standard UART, and the method further comprises the processing unit determining if a non-standard UART device is present.

35. *The method of Claim 33 further comprising:*

generating a series of digital values, in said processing unit; and

transmitting an analog signal based on the series of digital values, in a digital to analog converter, wherein said digital to analog converter is coupled to the processing unit by the bus.

Please cancel Claims 36 and 37 without prejudice or disclaimer.

38. *A computer comprising:*

a processing unit and memory programmed with a driver, said driver comprising (a) a software UART coupled to an operating system, (b) a software modem coupled to the software UART, and (c) an I/O handler coupled to the software modem; and

a device that does not comprise a UART, the device being coupled to the processing unit by a local bus, wherein the device comprises an analog to digital converter that generates sample digital values, and the device transfers the sampled digital values via the local bus to the software modem.

39. *The computer of Claim 38 wherein said device is hereinafter "first device" and said driver is hereinafter "first driver", the computer further comprising*

a second device comprising a UART, the second device being coupled to the processing unit by the local bus; and

a second driver comprising routines for accessing the second device.

40. *The computer of Claim 39 wherein:*

said first device is coupled to an I/O slot corresponding to a first COM port; and

said second device is coupled to an I/O slot corresponding to a second COM port.

41. *The computer of Claim 40 wherein:*

said first device occupies up to eight addresses on the local bus; and

said second device occupies eight addresses on the local bus.

42. *The computer of Claim 38 wherein:*

the memory is further programmed with routines for accessing another device that comprises a UART.

43. A computer comprising:

an analog to digital converter couplable to a communication medium to receive therefrom an analog communications signal and coupled to a local bus to transmit thereto a series of sampled digital values; and

a processor coupled to the local bus and programmed to:

determine data received based on a waveform represented by the sampled digital values and based on a modem protocol; and

provide the data received to an operating system of the computer.

44. The computer of Claim 43 wherein:

said processor is programmed to transmit a series of digital values on the local bus; and

the computer further comprises a digital to analog converter coupled to the local bus to receive therefrom the series of digital values and couplable to the communication medium to transmit thereto an analog signal based on the series of digital values.

45. The computer of Claim 44, wherein:

the analog to digital converter and the digital to analog converter are portions of a first device that does not comprise a standard UART;

the computer further comprises a second device that comprises a standard UART; and

said processor is programmed to:

access the first device through the operating system and a software UART; and

access the second device through the operating system and a standard COM driver.